INF4820: Algorithms for Artificial Intelligence and Natural Language Processing

Introduction and Overview

Stephan Oepen & Milen Kouylekov

Language Technology Group (LTG)

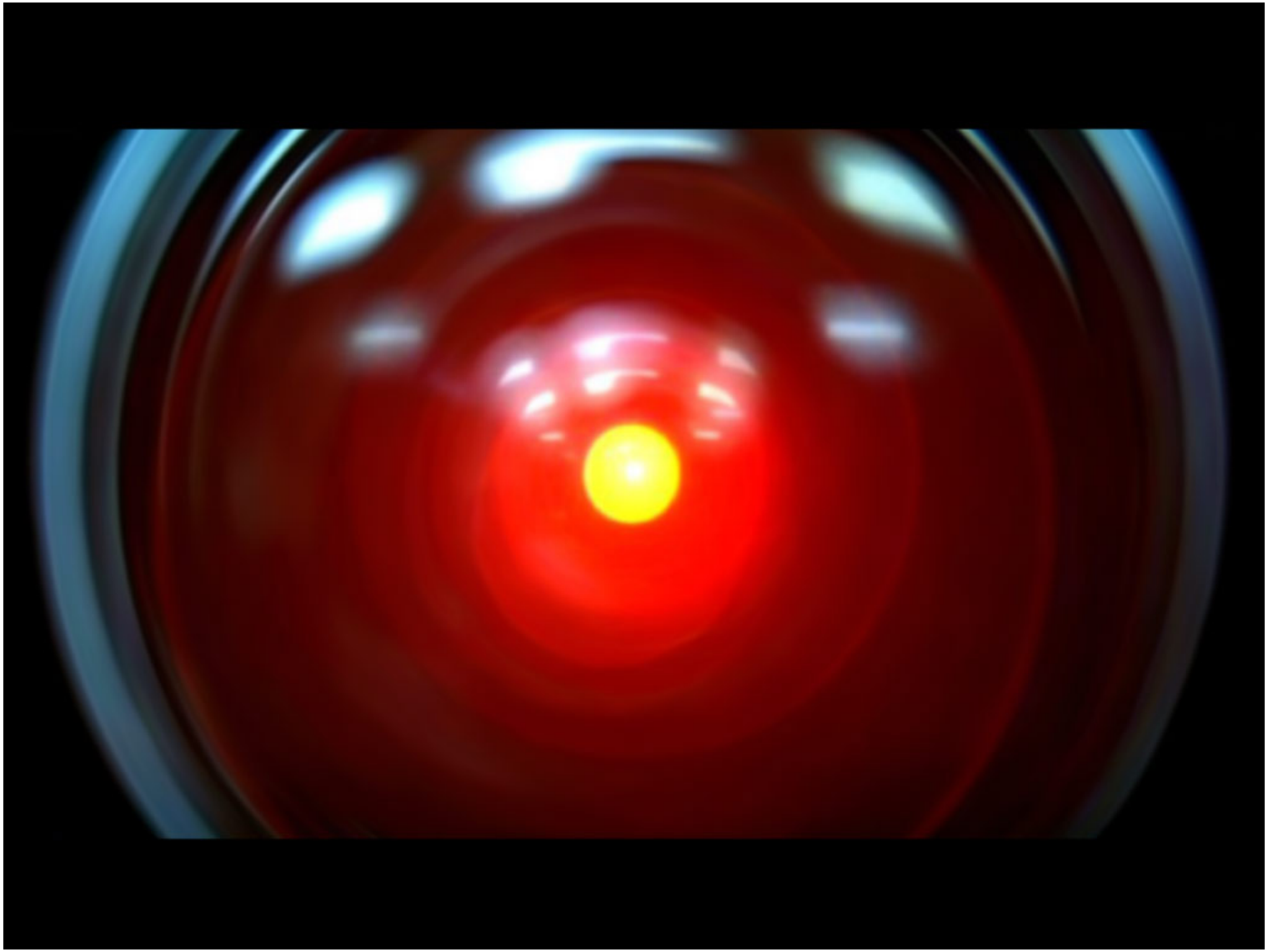
August 20, 2014

Topics for Today

Overview

- Course motivation and introduction:
- AI, NLP, ML — What are they?
- Lisp — What and why?
- Outline of lectures and learning goals.
- Practical details.

What is AI?



(HAL 9000 in *2001: A Space Odyssey*; 1968)

What is AI?

- Alan Turing, 1950:
  - *I propose to consider the question, 'Can machines think?'*
- The term 'AI' coined by John McCarthy (Dartmouth Conference, 1956).
  - *The science and engineering of making intelligent machines.*
  - *Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it.*
- Language always in central place, cf. the Turing Test.

What is AI? (cont'd)

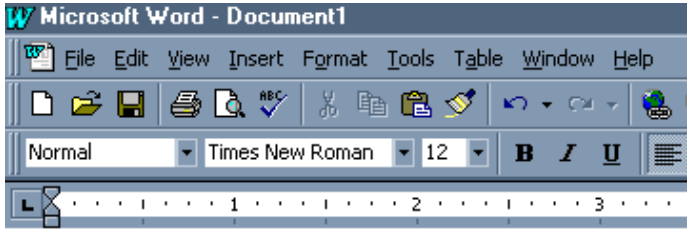
- The early years: simple chatbots, theorem proving, blocks world, expert systems, game playing (chess), ...
- Moving target: Whatever requires 'intelligent' decisions, but seems out of reach technologically?
  - Web search arguably would have been AI a couple of decades ago.
  - Open-domain Machine Translation out of reach until around 2005.
- For our purpose: AI is a toolkit of methods for problem solving and representation.

What is Natural Language Processing?

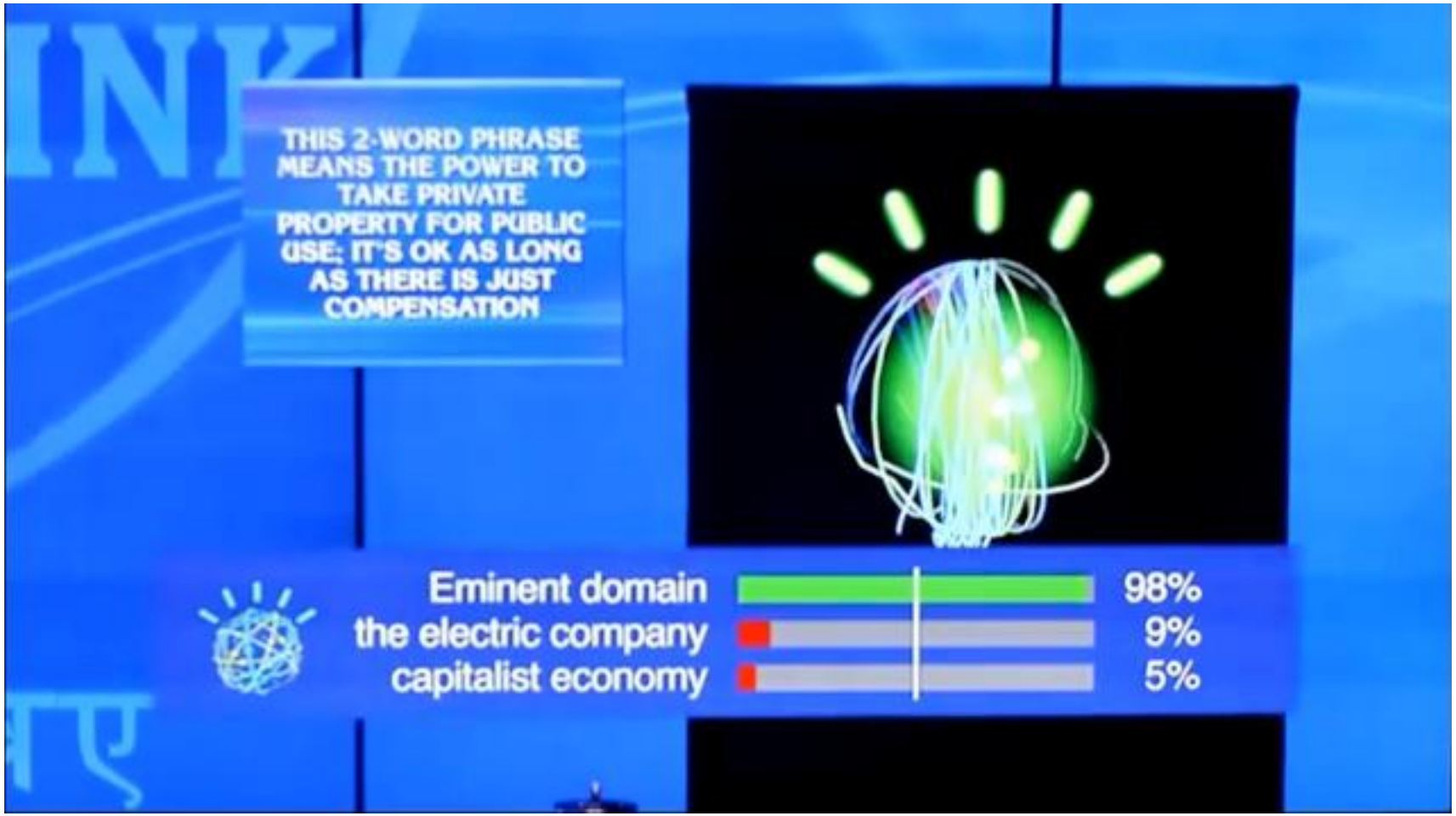
- Making computers 'understand' human language
- Aka language technology or computational linguistics
- Young and interdisciplinary field:
- Computer Science + Linguistics
- (+ Cognitive Science + Statistics + Information Theory + Machine Learning + . . .)

Some Applications

- Grammar and/or spell checkers, auto-completion
- Machine translation
- Q&A systems
- Dialog systems
- Speech recognition and synthesis
- Intelligent information extraction
- Summarization
- Sentiment analysis
- Any application requiring an understanding of language. . .



What is AI?



(IBM Watson beats long-time *Jeopardy!* champions; 2011)

What Makes NLP Hard?

Ambiguity

- I.e. the property of being open to multiple interpretations.
- All levels of linguistic description are associated with ambiguities.
- For humans, ambiguity is a feature: language is an efficient code.
  - The same expressions can be re-used in different contexts.
  - A large part of the information can be underspecified.
  - Interpretation relies on background knowledge and our expectations in a given context of use.
- Disambiguation is a central problem in NLP $\longrightarrow$ Search problems.

Ambiguity: Some examples

Word level ambiguity

- Norwegian: *rett*.
- English: *meal*, *dish*, *straight*, *correct*, *fair*, *justice*, *right*, *court*, *law*, *direct*, *grade*, . . . ?
- Ambiguous in meaning + syntactic category (part of speech).
- Need context to decide.

De hadde laget en deilig rett av grønnsaker.

Streken må være rett.

Kunden har alltid rett.

Du har rett til en advoktat.

Det er lovlig i henhold til norsk rett.

Slikt skjer rett som det er.

Vennligst rett disse prøvene!

Vi kjørte rett hjem.

Ambiguity: Some Examples

Referential Ambiguity

The authorities jailed the protesters because they { *advocated revolution.* / *feared revolution.*

Sentence-Level Ambiguity

I like eating sushi with { *tuna.* / *sticks.*

Acoustic Ambiguity

Let's talk about how to { *recognize speech* / *wreck a nice beach*

Some History: Different Approaches to AI and NLP

- Traditionally; two broad paradigms in NLP (and AI).
  - The rationalist approach, based on hand-crafted formal rules and manually encoded knowledge.
  - The empiricist approach, based on automatically inferring statistical patterns from data.
- 1950s – 80s: Rule-based
- Late 1980's: Empirical systems outperform rule-based in the area of speech recognition.
- 1990s: NLP as whole sees a shift of interest from rationalist towards empirical approaches.
- 2000s: No longer conceived as opposing poles, but complementary approaches typically used together.

The Basics of Empirical Methods

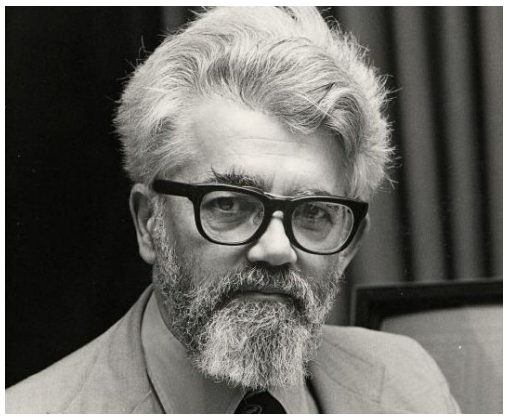
- The theoretical foundations are studied within the field of machine learning (ML) or statistical learning theory.

Machine Learning

... *the study of computer algorithms that improve automatically through experience* (Tom Mitchell 1997).

- Goal: Learn from examples, to make predictions about new data.
- Has applications in many other data-intensive sciences besides NLP, e.g. meteorology, biology, physics, robotics, signal processing, etc.
- An arsenal of methods: decision trees, support vector machines, maximum entropy models, naïve Bayes classifiers, artificial neural networks, genetic algorithms, ...

Lisp



- Powerful high-level language with long traditions.
- Especially strong support for symbolic and functional programming.
- "Discovered" by John McCarthy in 1958.
  - Initially intended as a mathematical formalism.
  - Then one of his students, Steve Russell, implemented an interpreter for the formalism, and Lisp the programming language was born.
- Rather than Lisp becoming outdated, the tendency has been that other languages have developed towards Lisp.

Common Lisp

```
(print "Hello world!")
```

- Several dialects; we will be using Common Lisp.
- Fully ANSI-standardized and stable.
- Rich language: multitude of built-in data types and operations.
- Easy to learn:
  - extremely simple syntax;
  - straightforward semantics.

An Experiment in Live Programming

The Factorial Function

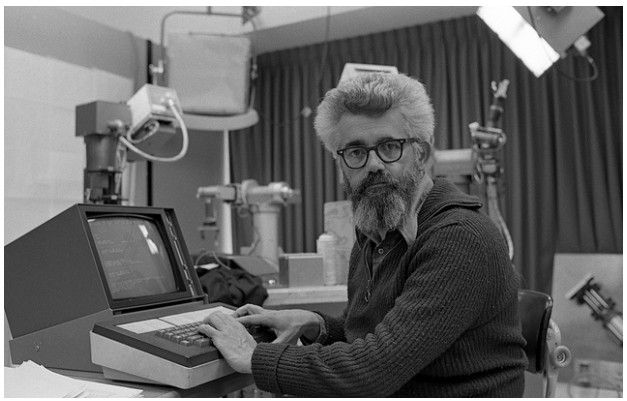
$$n! \equiv \begin{cases} 1 & \text{for } n = 0 \\ n \times (n-1)! & \text{for } n > 0 \end{cases}$$

Common Lisp Implementation

```
(defun ! (n)
  (if (= n 0)
    1
    (* n (! (- n 1)))))
```

A Note on Lisp and AI

- Often hailed (or dismissed) as "the AI language".
- While not quite true, there are several reasons for this coupling:
- AI coined by McCarthy in the mid-1950s.
- Lisp conceived by McCarthy in the mid-1950s.
- In addition to being fast and powerful, Lisp is particularly well suited for:
  - Explorative programming
  - Rapid prototyping
  - Incremental and interactive development
  - Extending the language itself

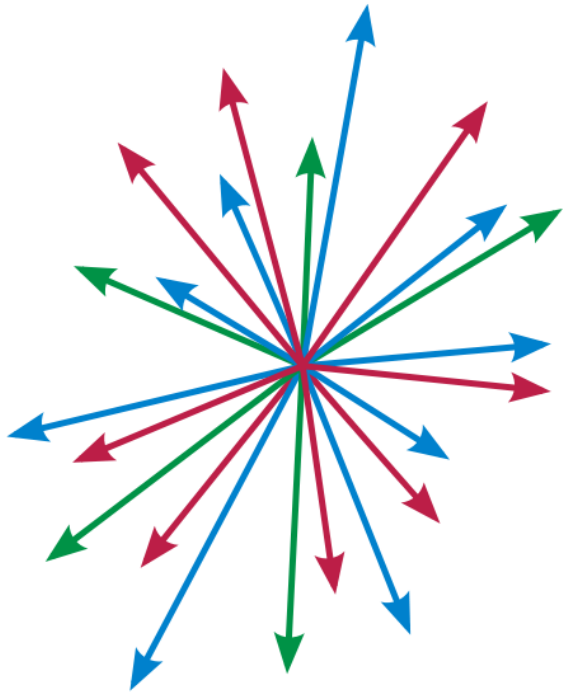


Lisp + Emacs = Good Match

- Steep learning curve, but with a big payoff:
- Emacs is an unusually powerful editor.
- Written in Emacs Lisp.
- Highly customizable—the Emacs Lisp dialect is also used as an extension language.
- Different “modes” make Emacs sensitive to different editing needs, e.g. depending on the specific programming language used.
- Prerequisite for an enjoyable Emacs experience: Spend some time mastering basic key commands!

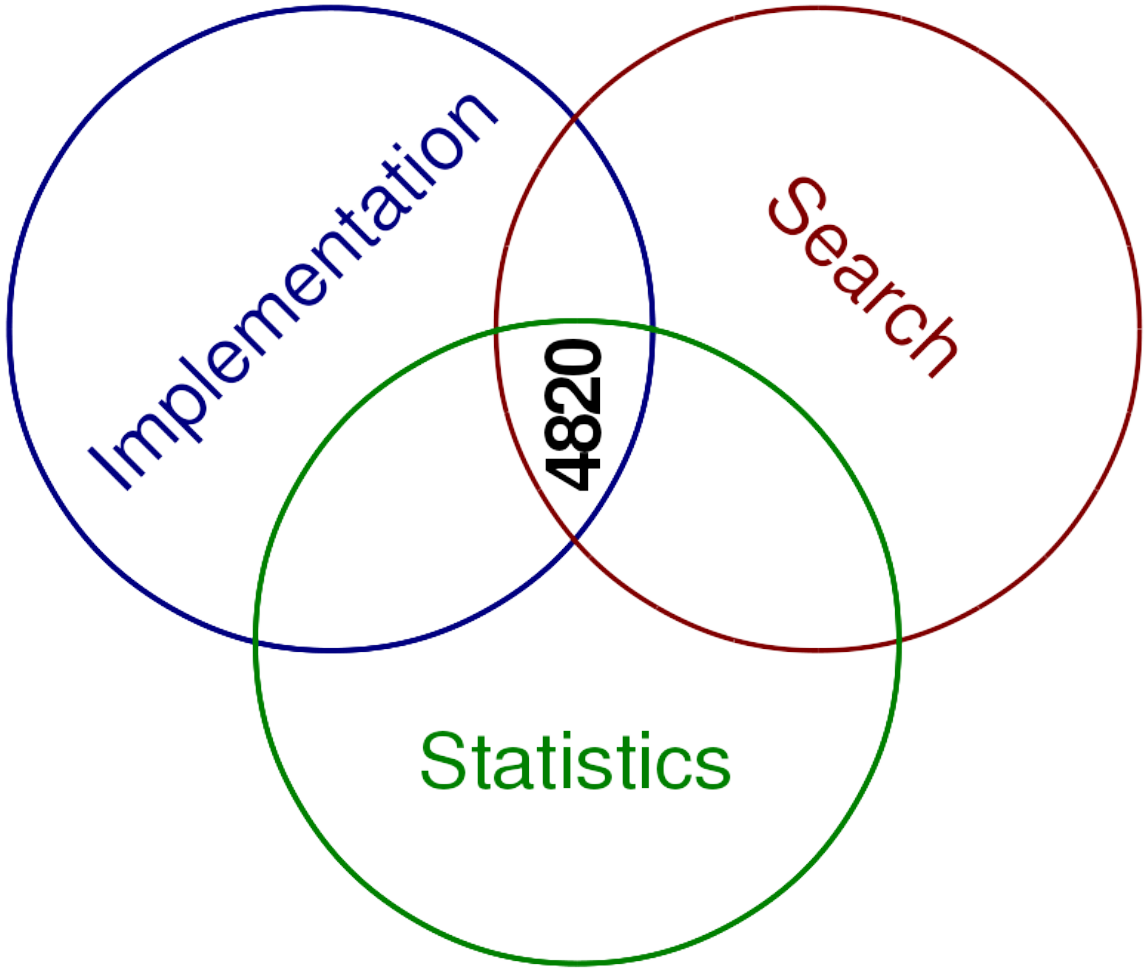
Overview of Lectures

- Common Lisp basics
- Vector space models
- Classification and clustering
- Probability theory
- Hidden Markov Models
- Statistical parsing



- Recurring themes: Machine learning, scalable data representations, search, dynamic programming.
- 4 hours of lectures, every other week + 2-hour laboratory weekly

Very High-Level Course Summary



Efficient and Scalable Algorithms and Data Structures for Searching (Probabilistically) Weighted Solution Spaces

Obligatory Exercises

- Three obligatory exercises:
- Exercise (2) and (3) have two parts each;
- Five problem sets in total.
- In order to pass and qualify for the exam you need a least
  - 6 of 10 possible points for Exercise (1),
  - 12 of 20 possible points for (2a) + (2b),
  - 12 of 20 possible points for (3a) + (3b).
- Extensions can only be given in case of illness, and re-submissions will not be possible.
- See course page for the schedule (tba).

An Experiment in High-Tech Teaching

- For student involvement and incremental exam preparation:
- occassional short quiz sessions $\rightarrow$ extra points towards exercises.

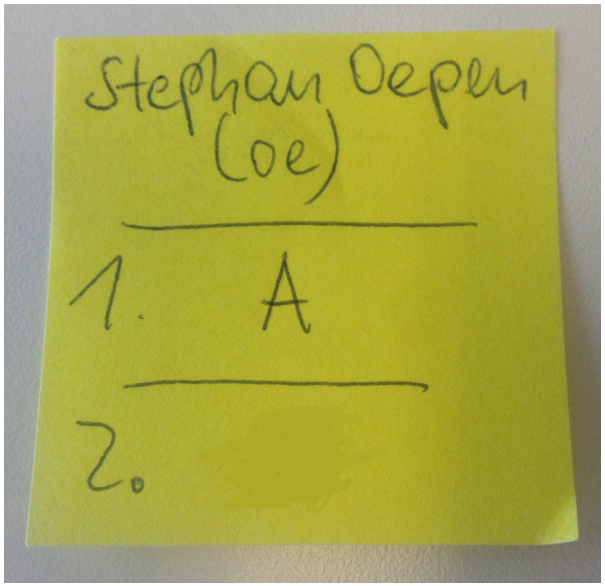
Example Quiz (0 + 0 Points)

1. Live programming can be useful?

   A: yes; B: no

2. Lisp was first developed by:

   A: Alan Turing; B: John McCarthy



Reading List

Obligatory reading; *selected parts* from:

- Jurafsky & Martin (2008):
  Speech and Language Processing (2nd Ed.)
- Seibel (2005):
  Practical Common Lisp (Available On-Line)
- Manning, Raghavan, & Schütze (2008):
  Introduction to Information Retrieval (Available On-Line)

Other recommended resources:

- Despite being 20 years old and long out-of-print *On Lisp* by Paul Graham is still a great read.
  - Freely available on-line: `http://www.paulgraham.com/onlisp.html`
- The Common Lisp 'HyperSpec':
  - `http://www.lispworks.com/documentation/HyperSpec/Front/`

Contact

- Questions?
  - On-line discussion board (*kursprat*) via course pages
  - `inf4820-help@ifi.uio.no` reaches all course staff:
  - Stephan Oepen
  - Milen Kouylekov
  - Jakob Tobias Frielingsdorf (laboratory assistant)
  - `{oe|milen|jakobtf}@ifi.uio.no`
- Messages:
  - Check your UiO email regularly;
  - Subscribe to the RSS feed of the course page;
  - Participate in the on-line discussion board.